DATA COMMUNICATION BETWEEN PROGRAMMABLE LOGIC CONTROLLERS IN THE INDUSTRIAL DISTRIBUTION APPLICATIONS

Anna BYSTRICANOVA[1], *Andrej RYBOVIC*[1]

[1]Department of Mechatronics and Electronics, Faculty of Electrical Engineering, University of Zilina, Univerzitna 1, 010 26 Zilina, Slovakia

anna.bystricanova@siemens.com, andrej.rybovic@fel.uniza.sk

Abstract. *The impact of automation is visible in all areas of industry as well as in everyday life. Automation makes the process control more efficient, increases productivity of work, manufacturing quality, decreases manufacturing costs. Automation is still in development so that it could succeed in filling all requirements of today's technical advance. For this reason we daily meet new questions about implementation of automation systems, their handling and expanding. One of these is the question of communication in industrial applications. In case of having more PLCs in one industrial network, it is necessary to solve their inter-communication. We should deal with this question in dependence on some facts, for example: used control system, used industrial network, transmission reliability requirements and so on. In this article we would like to present a solution for inter-communication between PLCs in one industrial network by S7 communication. S7 communication via Industrial Ethernet allows program-controlled communication using communication SFBs/FBs via configured S7 connections.*

Keywords

Automatization, Ethernet, CPU, CP, MPI, PLC, Profibus, RM-OSI.

1. Introduction

If the communication in the small applications is not critical than is sufficient to control these by one programmable logical controller (PLC). The role of the communication is exchanging of connection with a common computer in order to create and transmit the program to PLC and to transmit data to superior levels for operator's control of technology. At the direct control it is possible to rely on the response specified by manufactures between input change and adequate reaction (A-A´, Fig. 1). This response ranges about 10 ms depending on memory size, processor speed or on the longest time of programming repetition cycle (scan time). While building distribution control system for controlling large industrial application it is very important to solve the question of communication. Sometimes there arises the question about controlling output on the remote PLC, which is available only via industrial communication network (B-B´, Fig. 1).

In this case the response will be bigger and depend on a lot of factors. Industrial networks cover mainly production control, so it is very important to ensure high reliability, deterministic mode of communication and high power. On the other hand industrial networks enable connection (C-C´, Fig. 1) with centralized operator layer (PC with function SCADA/MES) of whole company information system and top company ERP system. Common corporation (proprietary) industrial networks use only three layers from standard reference communication model (RM-OSI) – physical, link and application layer. See comparison in the Tab. 1. Modern networks based on industrial Ethernet and TCP/IP technologies use five layers, where network layer and transport layer are allocated to physical and link layers to support TCP/IP technology [1].

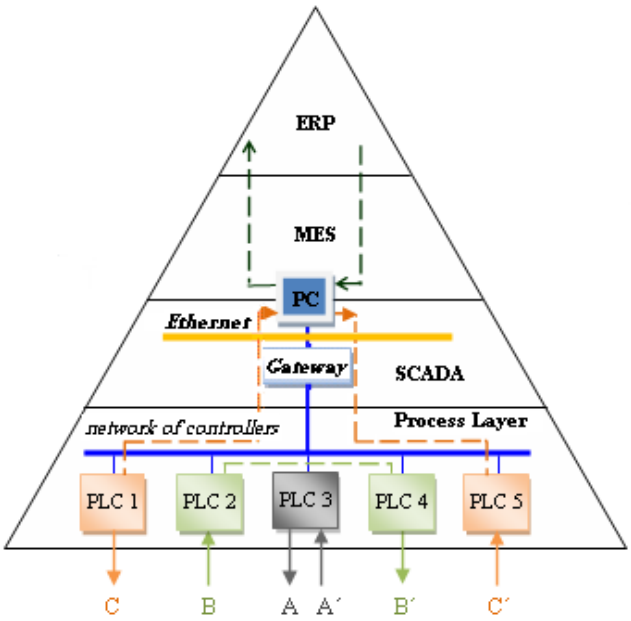

Fig. 1: Business information model.

2. Basic Questions

Managing connections between remote PLC performs the data link layer with use of transmission in real time. Total segmentation of current industrial networking protocols is expressed in Tab. 1 [1].

Tab.1: Comparison of communication models of industrial networks with a reference model RM-OSI.

Classic Network		RM-OSI	Industrial Ethernet TCP/IP
Application	7	Application	Application
	6	Presentation	
	5	Session	
	4	Transport	TCP/UDP
	3	Network	IP
CAN, MAC, LLC	2	Data Link	MAC, LLC
CAN, Profibus....	1	Physical	IEEE 802.3, 4

Communication Gateway is used to link network controllers and Ethernet.

When establishing a communication model at level of programmable logic controllers, we face to fundamental issues of network selection with given interface and protocol with regard to the used hardware.

Which network to choose?

Services of industry computer networks have primarily been designed to support process control in real time and for general automation applications. Services of industrial networks have been specifically designed, particularly with regard to needs of individual manufacturers of automation technologies and systems. Therefore their standardization was delayed and only after a certain period of time it is possible to monitor profiling of typical application service of industrial networks [1].

What type of protocol / service to select?

In order for the PLC to be able to know how to communicate in common network, they use the same communication protocol. It is actually a summary of parameters and rules governing communication. Selection of the communication protocol depends on selection of network.

2.1 Basic Network Characteristics

The architecture of the Simatic Siemens-Net was designed with intention of creating an integrated environment for open communication of various automation systems (Total Integrated Automation) at all levels of industrial systems. Architecture of these networks integrates various network technologies so they ensure providing required communication services. Within the architecture are defined the following hierarchy levels with designated mode of communication:

- IT communication – allows integration of automation equipment with Enterprise Information System.
- Data communication – allows to communicate in real time on procedural level and control level, in which communication of PLCs, programming and controlling of performance components are dominant.
- Procedural communication – allows I/O operations in real time and communication with sensors and sensor manufacturing process [1].

As previously mentioned, in this case we are interested in data communication within industrial networks Simatic-Net, where we characterize basic types of industrial networks.

1) MPI – Multi Point Interface

MPI bus is designed for programming and data services on devices. It is not designed to collect data from decentralized peripherals. In a network there must be at least one Master, which manages data flow on the network. Network speed is optional, 9 kbit/s to 12 Mbit/s. In principle, transmission distance is not limited, but the network is primarily intended for local road length in tens to hundreds meters.

Transmission technology is addressed through the RS485 communication standard transmission and by

fiber optic supplemented with converters, but this solution is not commonly used.

2) Disadvantages of Using:

- limited amount of data transferred,
- longer response time,
- short range network.

PLCs from other manufacturers such as Siemens do not usually have MPI interface [2], [3].

3) Profibus–Proces Field Bus

Is an industrial fieldbus used for all areas of automation. At the level of physical layer Profibus is specified to suit the diverse needs and to support majority of industrial applications, such as linking remote controller peripherals to procedural controller (DP) or procedural automation (PA) [1].

4) Types and Versions of Profibus

Profibus DP (Decentralized Periphery). This is the simplest and highly predominant variant of Profibus. It is suitable for fast transfer of decentralized peripherals and remote I/O units. Communication medium is either a twisted pair (standard RS-485) or optical fiber at speed of up to 12 Mbit/s.

Profibus PA (Proces Automation) uses enlarged standard Profibus DP and is designed to control slow processes, especially in potentially explosive atmospheres as it corresponds to intrinsical spark safety.

Profibus FMS Profibus FMS provides communication standard for communication in heterogeneous environment with a large set of services for working with data, programs and alarms. Communication medium is a twisted pair (standard RS-485) or optical fiber. Speed is lower than the Profibus DP [1].

5) Disadvantages of Using

Comparing to Ethernet, Profibus is less powerful and flexible network.

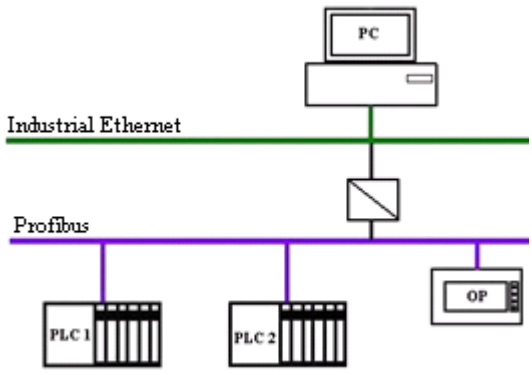

Fig. 2: System architecture network Simatic – Net.

6) Industrial Ethernet

Industrial Ethernet is based on standard IEEE 802.3. This protocol defines the physical layer and data link layer model. As a consequence, the standard IEEE 802.3 specifies characteristics of communication interface and a method of managing the access to transmission medium [1].

Systems with industrial Ethernet can have different structures, i.e. network topology, logical structure of communication links but also methods of transmitting data. Topology: bus, tree, star, ring [3].

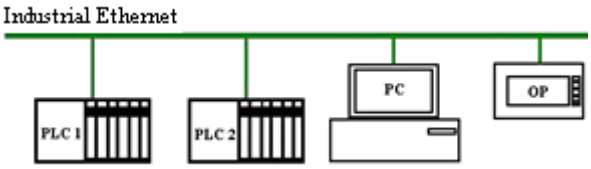

Fig. 3: Bus topology Ethernet network.

In the process of development and use of Ethernet in industrial practice were created many solutions. International Electrotechnical Commission (IEC) has recognized following Ethernet protocols:

- Profinet IO – cyclic communication addressing through MAC (Device Name),
- ISO – acyclic communication addressing through MAC,
- ISO-on-TCP – acyclic communication addressing through IP,
- TCP/IP – reliable acyclic confirmed by communication addressing through IP,
- UDP/IP – acyclic datagram by uncertified communication addressing through IP [1].

3. Communication among Several PLC

Managing connections between remote PLC performs the data link layer with use.

3.1 Creating Project of Communication between PLC

For practical verification of communication among various PLC, we used the communication model consisting of the following parts.

Hardware:

- Simatic S7-300 (CPU 315-2 PN/DP),
- Simatic S7-300 (CPU 315-2 PN/DP),

Software:

- Step 7 S7/M7/C7 (Version V5.4 + SP5 + HF1).

Network:

- Ethernet.

Protocol:

- S7 communication (with systemic blocks BSEND, BRCV).

The Ethernet CP for SIMATIC S7 supports this type of communication depending on the CP type. S7 communication forms a simple and efficient interface between SIMATIC S7 stations and PGs/PCs using communication function blocks.

In Simatic Manager, we have created a new project, into which two SIMATIC 300 stations were inserted. The project is marked 1 and 2.

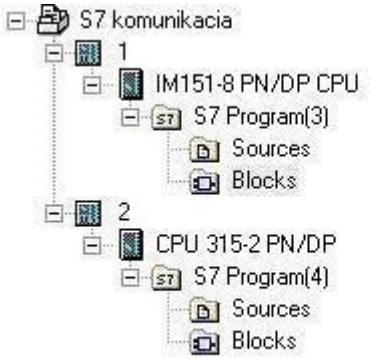

Fig. 4: The project created in SIMATIC Manager environment.

Into the part Hardware Programming Environment Step 7 we insert a specific type of processor (CPU) for each PLC. For each processor an IP address is be set.

In Step 7 we open environment for the Net, where we create a connection between the communication partners in the following way:

- Menu- Insert- New Connection.

We choose the communication partner within one project. Type of connection is: S7 connection.

The following picture illustrates the characteristics of created connection between two CPU.

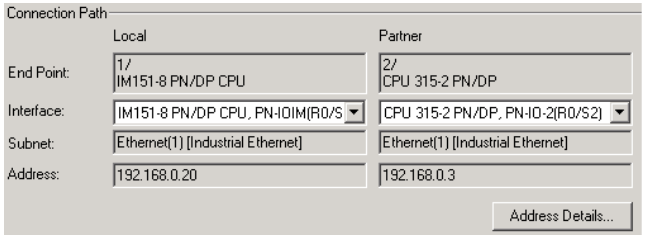

Fig. 5: Properties of S7 connection.

3.2 PLC Program

After setting communication between processors (CPUs) of the both PLCs, we could access to creation of the user program. It is possible to choose from the following programming languages in programming environment of Step 7:

- STL (*Statement List Programming Language*): a text-based programming language with a structure similar to machine code. Each statement represents a program processing operation of the CPU,
- LAD (*Ladder Logic*): a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay-based logic hardware,
- FBD (*Function Block Diagram*) a graphical language that allows the user to program elements in "blocks".

These program languages are equivalent. We could change them while programming. User program structure is declared by organization block (OB 1), which operates cyclically. Operation system S7 CPU periodically initializes the block OB1. OB1 consists of separately programmable function blocks (FBs). FB contains memory, which makes it possible to save internal variables to this block by declaration table. For the most frequently used routines we use function – FC blocks. Data blocks (DBs) are used for saving user data, which could but need not to couple with particular function block FB. Program Step 7 uses System Function Blocks SFBs and System Functions SFCs. These are directly integrated in the S7 processor (CPU) and enable entry to some special system functions. It is necessary to call programmed blocks FB, FC, DB by appropriate organization block (OB) [5].

Into each PLC programme we put shared Data Blocks, which are not associated with a concrete FB block. We mark them as DB200 and DB 201.

DB 200: DB contains data sent from one PLC to another.

DB2001: DB contains data received from the other PLC.

Size and data type in Data Blocks are visible from the pictures.

Address	Name	Type
0.0		STRUCT
+0.0	DEI	ARRAY[0..2]
*2.0		INT
+6.0	AEI	ARRAY[0..10]
*4.0		REAL
=50.0		END_STRUCT

Fig. 6: Data Block DB200 sending data.

Address	Name	Type
0.0		STRUCT
+0.0	DEI	ARRAY[0..2]
*2.0		INT
+6.0	AEI	ARRAY[0..10]
*4.0		REAL
=50.0		END_STRUCT

Fig. 7: Data Block DB201 receiving data.

We create Function Block FB1 into which we put blocks SFB 12 and SFB 13.

From the program library we choose system functions SFB 12 BSEND and SFB 13 BRCV. Function block FB1 containing functions for data transfer is periodically operated at the main block OB1 together with its data block DB1.

1) Description of the System Block SFB12/FB 12

SFB/FB 12 "BRCV" sends data to a remote node (partner), SFB/FB, which is a type of "BRCV". The data area to be transmitted is segmented. Each segment is sent individually. Last segment, when adopted confirmed partner. In this way data transmission may be transmitted between communication partners more than other communication SFB/FB blocks configured for connection to the S7 65534 Bytes through an integrated interface.

S7-300: Sending data is activate with help leading edge input REQ. The parameters R_ID, ID, SD_1 and LEN are transferred on each positive edge at REQ. After a job has been completed, you can assign new values to the R_ID, ID, SD_1 and LEN parameters. To carry out the transfer of the segment data block that must be called periodically in the user program. The starting address and the maximum length of data sent is specified in SD_1. In the input parameter LEN defines the size of sending data field in bytes [4].

2) Description of System Block SFB13/FB 13

SFB/FB 13 "BRCV" receive data from a remote partner SFB/FB, which is a type of "BSEND. After each received data segment sent a confirmation partner SFB/FB and LEN parameter updating [4].

4. Communication among Several PLC

After programming system blocks and following filling CPUs of both PLC, we can make check communication procedure between two PLCs. Block BSEND sends data to the other PLC, where the block BRCV receives it. Settings system of the blocks is clear from the pictures below.

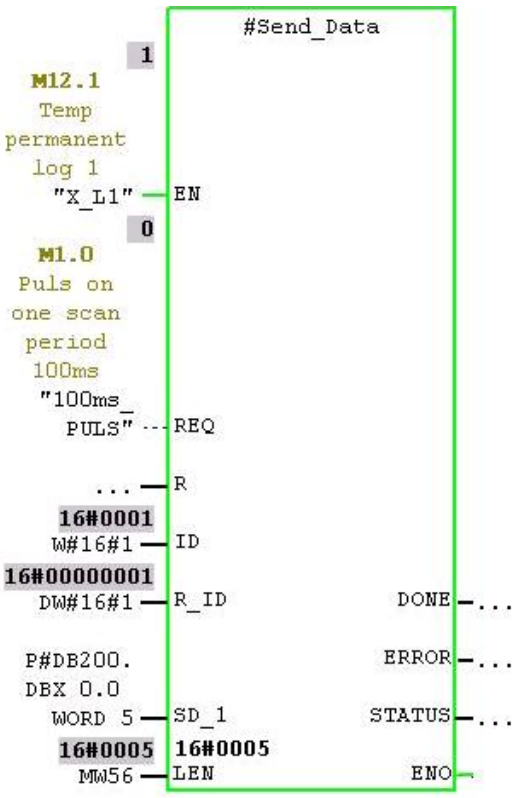

Fig. 8: Block SFB 12 BSEND.

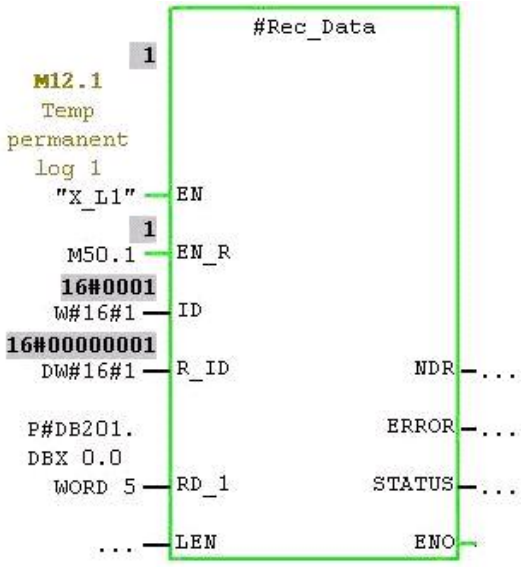

Fig. 9: Block SFB 13 BRCV.

Mutual communication was verified by VAT tables. Data sent from DB 200 via BSEND function from PLC 1 were received by the other PLC via function BRCV into DB 201. Data could be sent and received separately by particular elementary types (bit, byte, word, int, real,..) or by whole data areas (array, ...).

REC DATA -- @S7 komunikacia\1\IM151-8 PN/DP CPU\S7 Program(3) ONLINE

	Address	Symbol	Display format	Status value
1	//receive data			
2	DB201.DBX 0.0		BOOL	true
3	DB201.DBX 0.1		BOOL	true
4	DB201.DBX 0.2		BOOL	true
5	DB201.DBX 0.3		BOOL	true
6	DB201.DBX 0.4		BOOL	true
7	DB201.DBX 0.5		BOOL	true
8	DB201.DBX 0.6		BOOL	true
9	DB201.DBX 0.7		BOOL	true
10	DB201.DBD 6	"Data_1<-2".AEI[0]	FLOATING_POINT	20.0
11				
12	//send data			
13	DB200.DBX 0.0		BOOL	false
14	DB200.DBX 0.1		BOOL	false
15	DB200.DBX 0.2		BOOL	false
16	DB200.DBX 0.3		BOOL	false
17	DB200.DBX 0.4		BOOL	false
18	DB200.DBX 0.5		BOOL	false
19	DB200.DBX 0.6		BOOL	false
20	DB200.DBX 0.7		BOOL	false
21	DB200.DBD 6	"Data_1->2".AEI[0]	FLOATING_POINT	77.0
22	DB200.DBX 1.0		BOOL	true

Fig. 10: VAT table in PLC 1.

BSEND, BRCV -- @S7 komunikacia\2\CPU 315-2 PN/DP\S7 Program(4) ONLINE

	Address	Symbol	Display format	Status value
1	//receive data			
2	DB201.DBX 0.0	"Data_1<-2".status	BOOL	false
3	DB201.DBX 0.1	"Data_1<-2".done	BOOL	false
4	DB201.DBX 0.2	"Data_1<-2".error	BOOL	false
5	DB201.DBX 0.3	"Data_1<-2".rez1	BOOL	false
6	DB201.DBX 0.4	"Data_1<-2".rez2	BOOL	false
7	DB201.DBX 0.5	"Data_1<-2".rez3	BOOL	false
8	DB201.DBX 0.6	"Data_1<-2".rez4	BOOL	false
9	DB201.DBX 0.7	"Data_1<-2".rez5	BOOL	false
10	DB201.DBD 6	"Data_1<-2".word3.analog1	FLOATING_POINT	77.0
11	DB201.DBX 1.0		BOOL	true
12	//send data			
13	DB200.DBX 0.0		BOOL	true
14	DB200.DBX 0.1		BOOL	true
15	DB200.DBX 0.2		BOOL	true
16	DB200.DBX 0.3		BOOL	true
17	DB200.DBX 0.4		BOOL	true
18	DB200.DBX 0.5		BOOL	true
19	DB200.DBX 0.6		BOOL	true
20	DB200.DBX 0.7		BOOL	true
21	DB200.DBD 6	"Data_2->1".AEI[0]	FLOATING_POINT	20.0

Fig. 11: VAT table in PLC 2.

5. Achieved Experimental Results of the Project

At the end we carried out the experiment when we were in one minute tracking the number of cycles of data transferred between PLC at:

- change impulse length of parameter REQ in block BSEND,
- change of capacity of transferred data.

We followed one bit from the transferred data block. Its state was changing (0,1) every realized cycle of transmission of data during one minute. We used counter C1, which counted states of variable in 1 and counter C2, which counted states of variable in 0.

We found out that in size 10 B of transferred data and time 100 ms in variable REQ of BSEND in both PLCs 578 cycles data communication during 1 minutes were executed at scan times CPU 1 (Fig. 12) and CPU 2 (Fig. 13).

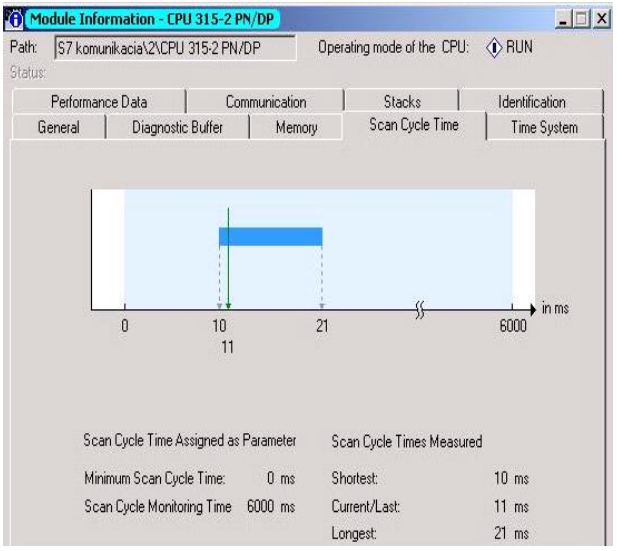

Fig. 12: Scan time CPU 1.

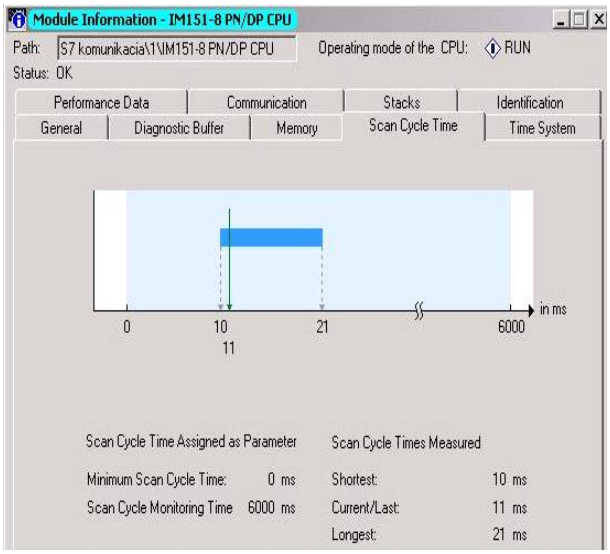

Fig. 13: Scan time CPU 2.

C	1		COUNTER	C#289
C	2		COUNTER	C#289

Fig. 14: Cycle counters of transferred data with size 10 B and 100 ms impulse.

We carried out the same experiment at 1 s impulse in parameter REQ of BSEND block. In this case 60 cycles between two PLCs during 1 minute were realised.

It means that at 10 times decreased frequency of data communication between system blocks of both PLCs, the number of realised cycles of data transfer decreased almost by 10 times. There was not a shining changing scan time both CPUs. No significant change in scan time was registered.

Fig. 15: Counter cycles of transferred data with size 10 B a 1 s impulse.

This time we increase the capacity of communicated data on maximum, i.e. 32 kB. At the impulse length of 100 ms the transfer was executed 338 times during 1 minute. Scan time was the same for both CPUs.

Fig. 16: Counter cycles of transferred data with size 32 kB and 100 ms impulse.

We tried to decrease frequency of transfer. To the parameter REQ of BSEND blocks of both PLCs we connected 1s impulse, which inicialized sending data from one PLC to the other. In this case transfer was executed 26 times only.

Fig. 17: Counter cycles of transferred data with size 32 kB and 1 ms impulse.

The result of these experiments is that scan time of both processors was considerably changed neither with condensed transfer frequency nor with bigger capacity of communication data. It means that the examined data communication was executed reliably and did not load the processors of both logical automats.

6. Conclusion

By the practical example one form of data communication among more PLCs was demonstrated. We chose the model where both processors were the SIMATIC type. We chose the Industrial Ethernet network with service S7 communication. For data transfer we used system blocks SFB 12 BSEND and SFB 13 BRCV, which are able to transfer maximum data capacity 32 kB for S7 – 300 and 64 kB for S7-400.

As it has been mentioned and experimentally verified, the data communication speed among several PLCs mainly depends on used hardware (CPU type), industrial network used for data transfer and parameter settings of system blocks including the size of transferred data.

Acknowledgements

This article was written with the support of a guaranteed project APVV VMSP-P-0085-09.

References

[1] FRANEKOVÁ, M.; KÁLLAY, F.; PENIAK, P.; VESTENICKÝ, P. *Komunikačná bezpečnosť priemyselných sietí*. Žilina 2007. ISBN 9078-80-8070-715-6.

[2] BÉLAI, I. *Industrial communication : Lecture* [online]. 2007, 18. 1. 2007. Available at WWW: <http://www.kar.elf.stuba.sk/predmety/pkom/PKS/Prednasky/pks_prednasky.html>.

[3] KOSEK, R.; VOJANEC, J. *Novinky a komunikace SIMATIC* [online]. 27. 1. 2010. Available at WWW: <http://www.siemens.cz/siemjetstorage/files/57032_01$komunikace.prehled.pdf>.

[4] SIEMENS AG. *Help on Simatic Manager V5.4+SP5+HF1*. Siemens AG, 1995-2009.

[5] ONDROVIČOVÁ, M. *Control system SIMATIC S7-300* [online]. 20. 1. 2010. Available at WWW: <http://www.kirp.chtf.stuba.sk/~ondrovic/simatic.htm>.

About Authors

Anna BYSTRICANOVA was born in 1983 in Trencin. She received her master`s degree in the field of study „Automatization“ at the University of Zilina - Faculty of Mechanical Engineering - Department of Machining and Automatization. Nowadays she works as PLC programmer. She is an external Ph.D. student at the University of Zilina - Faculty of Electrical Engineering - Department of Mechatronics and Electronics in the study programme „Automatization“.

Andrej RYBOVIC was born in 1984 in Hrusvic (Slovakia). He received her master`s degree in the field of study „Power Electronics Systems“ at the University of Zilina - Faculty of Electrical Engineering - Department of Mechatronics and Electronics. Nowadays he is a Ph.D. student at the University of Zilina - Faculty of Electrical Engineering - Department of Mechatronics and Electronics in the study programme "Power Electronics Systems". He is about design of mechatronics models with hardware Loop Back (HIL).